**Name: Angad Sandhu**

**Registration Number: 190905494**

**Section: CSE-A**

**Roll Number: 60**

**Subject: DS Lab 5**

**1) Try the given wordcount program for heart disease dataset, covid 19 dataset, example dataset and german credit dataset**

**# mapper.py**

import sys

for line in sys.stdin:

words = line.strip().split(',')

for word in words:

print("%s\t%d"%(word, 1))

**#reduce.py**

from operator import itemgetter

import sys

current\_word = None

current\_count = 0

word = None

for line in sys.stdin:

try:

word, count = line.strip().split('\t', 1)

count = int(count)

except ValueError:

continue

if current\_word == word:

current\_count += count

else:

if current\_word:

print("%s\t%d"%(current\_word, current\_count))

current\_count = count

current\_word = word

if current\_word == word:

print("%s\t%d"%(current\_word, current\_count))

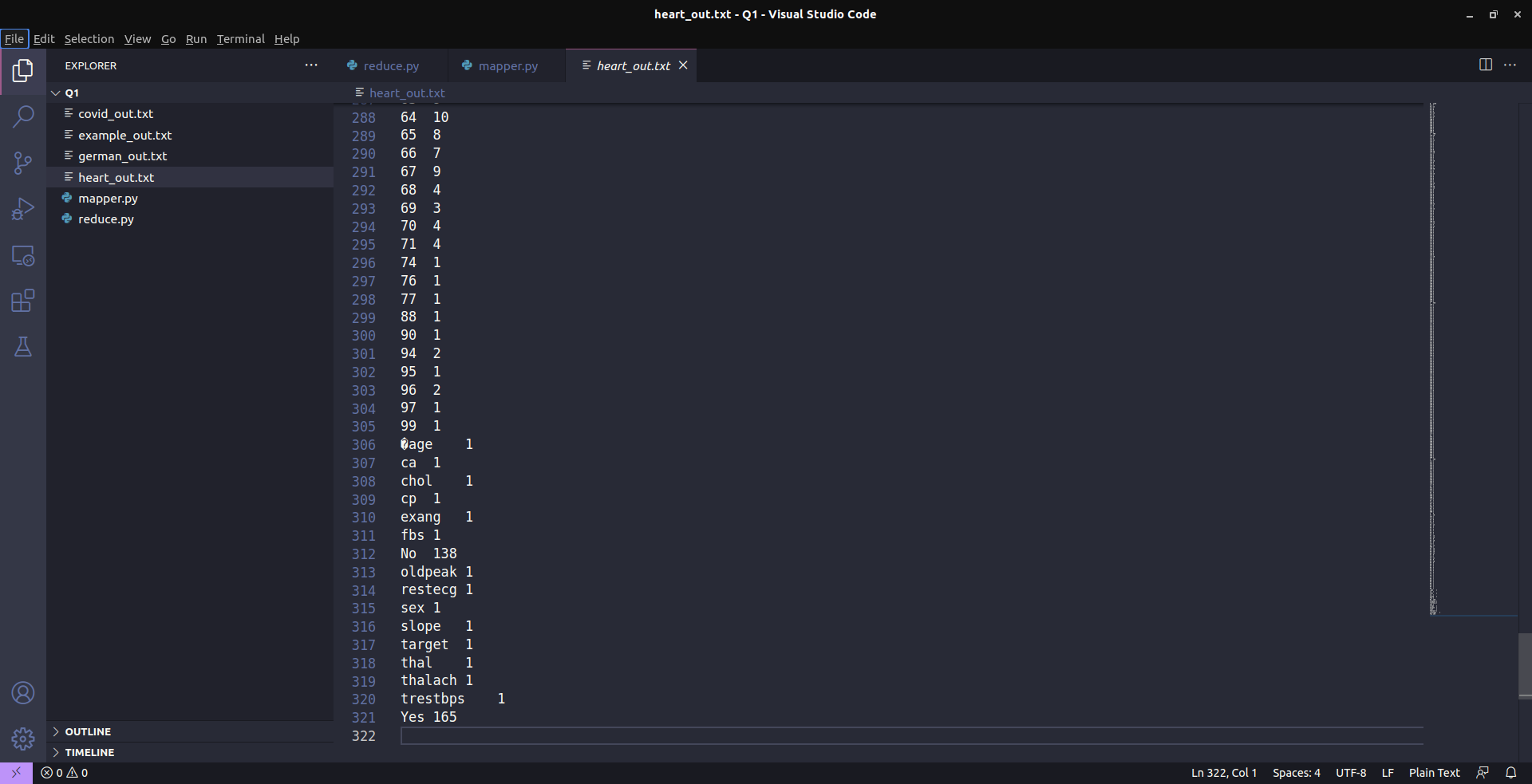
**Output:**

Text

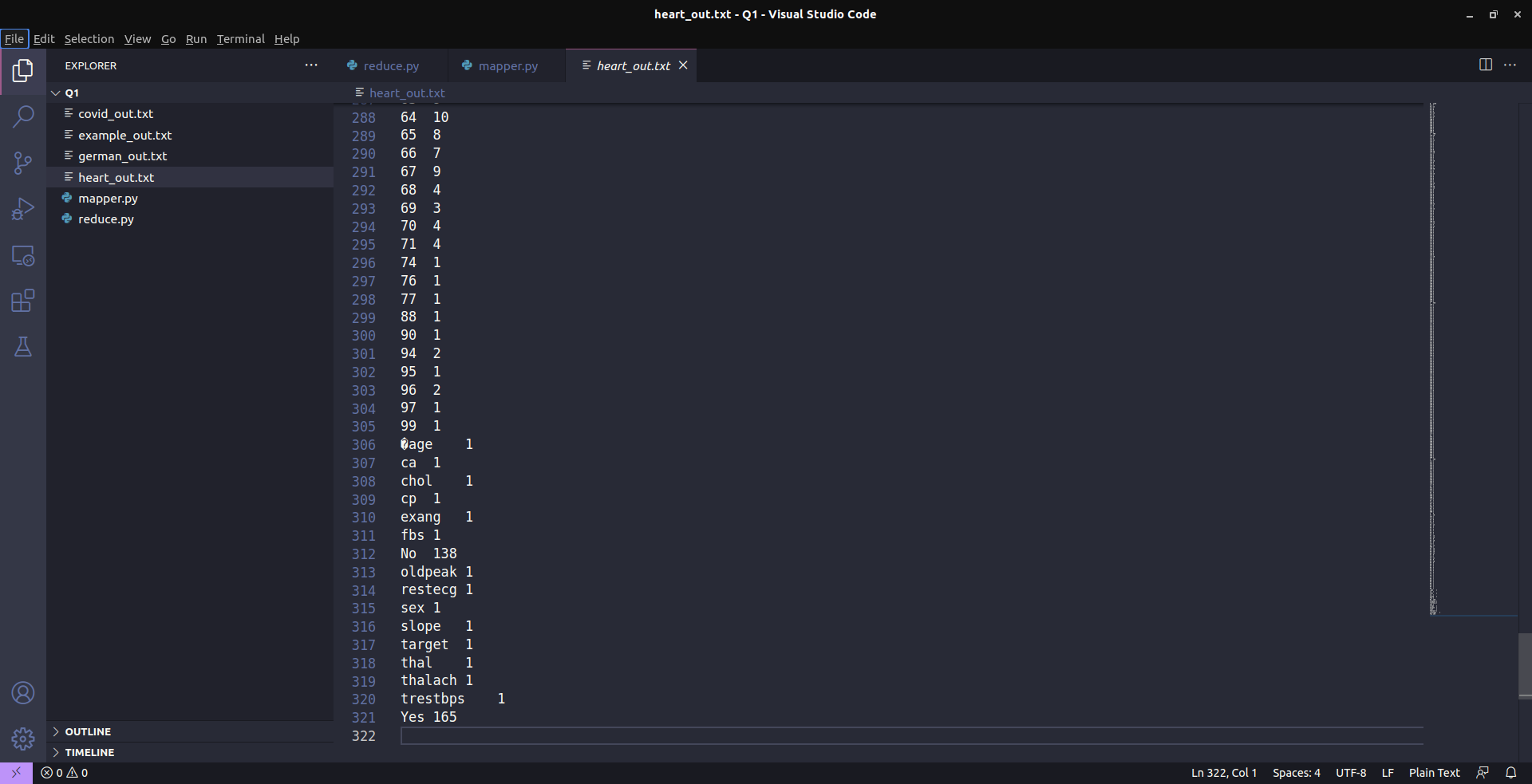
Description automatically generated

**Output files (trucated)**

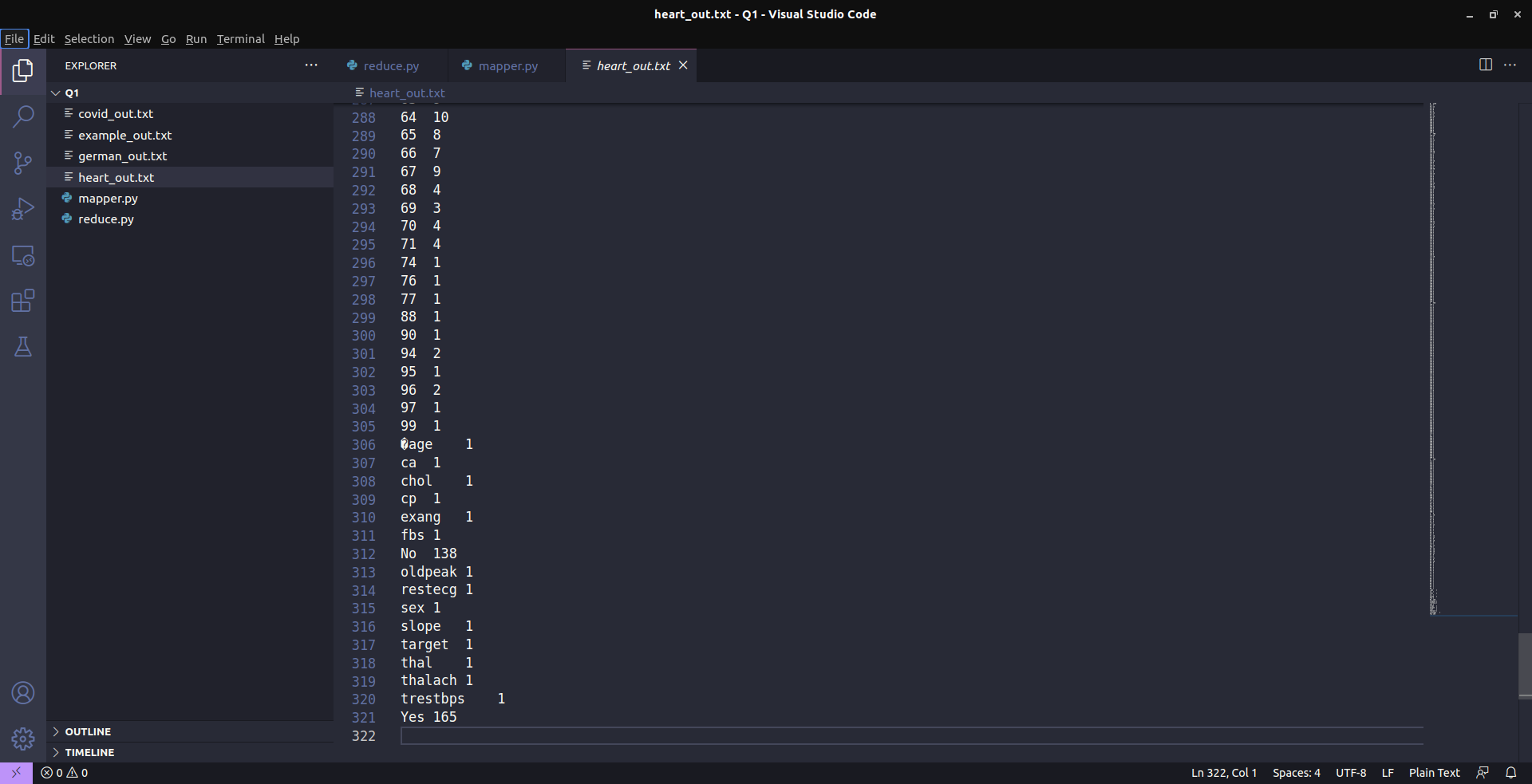
**heart\_out.txt:**

****

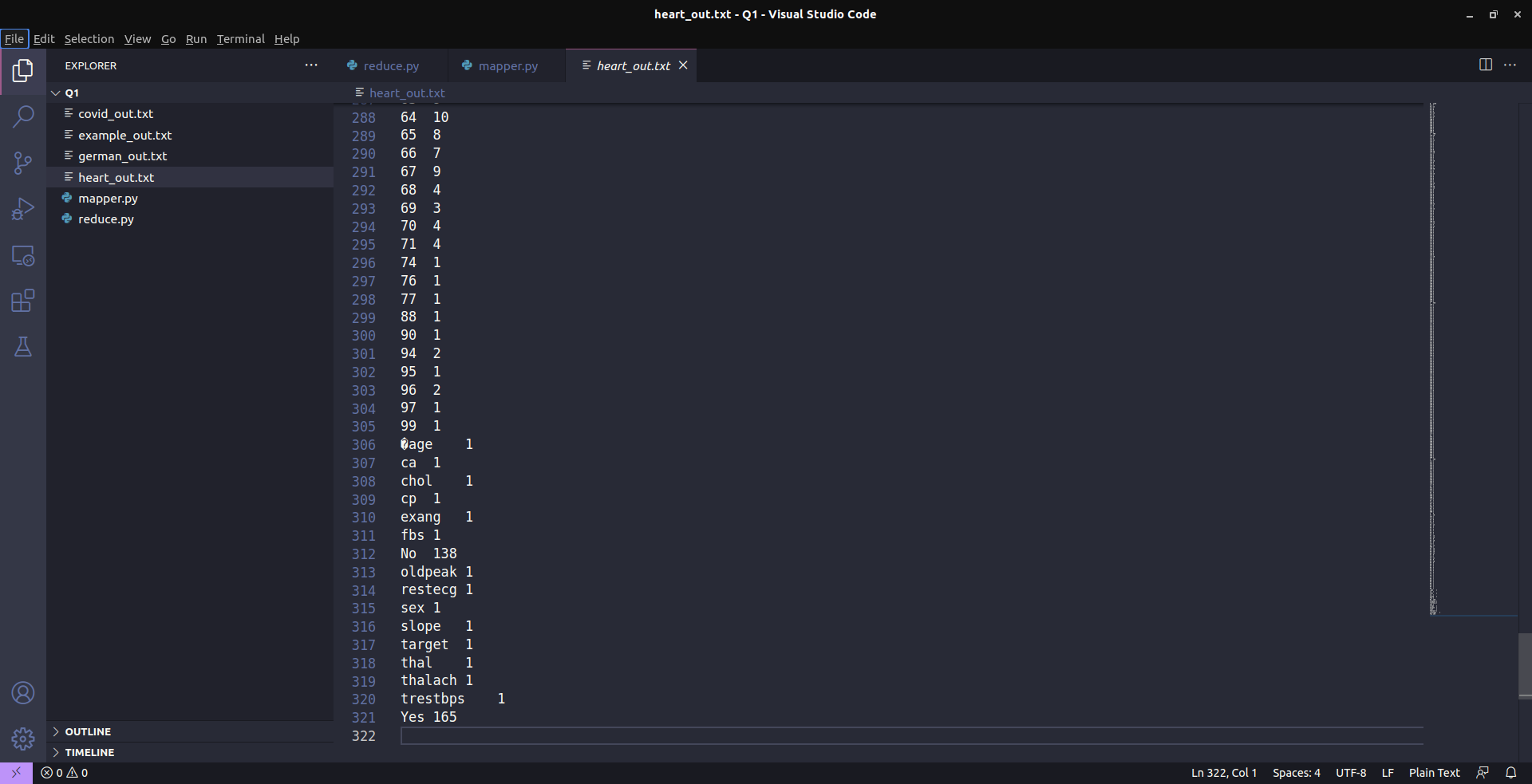
**covid\_out.txt:**

****

**example\_out.txt:**

****

**german\_out.txt:**

****

**2) Map Reduce Program to find frequent words**

**Try the given frequent word count program for heart disease dataset, covid 19 dataset, example dataset, and german credit dataset**

**#freqmap1.py:**

import sys

for line in sys.stdin:

L = [ (word.strip().lower(), 1) for word in line.strip().split(',') ]

for word, n in L:

print("{}\t{}".format(word, n))

**#freqred1.py**

import sys

lastWord = None

sum = 0

for line in sys.stdin:

try:

word, count = line.strip().split('\t', 1)

count = int(count)

except ValueError:

continue

if lastWord == None:

lastWord = word

sum = count

continue

if word == lastWord:

sum += count

else:

print("{}\t{}".format(lastWord, sum))

sum = count

lastWord = word

# output last word

if lastWord == word:

print("{}\t{}".format(lastWord, sum))

**#freqmap2.py:**

import sys

for line in sys.stdin:

word, count = line.strip().split('\t', 1)

count = int(count)

print("{}\t{}".format(word, count))

**#freqred2.py:**

import sys

mostFreq = []

currentMax = -1

for line in sys.stdin:

word, count = line.strip().split('\t', 1)

count = int(count)

if count > currentMax:

currentMax = count

mostFreq = [ word ]

elif count == currentMax:

mostFreq.append(word)

for word in mostFreq:

print("{}\t{}".format(word, currentMax))

**Output:**

Text

Description automatically generated

**heart\_out.txt:**

**0 1145**

**covid\_out.txt:**

**0 45012**

**example\_out.txt:**

**amex 13**

**german\_out.txt:**

**1 700**

**3) MapReduce Program to explore the dataset and perform filtering (typically creating key value pairs) by mapper and perform count and summary operations on instances.**

**#itemmap.py (for heart disease dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 14:

age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, target = data

print("{}\t{}".format(age, trestbps))

**#itemmap.py (for covid 19 dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 8:

sno, observationdate, province, country, lastupdate, confirmed, deaths, recovered = data

print("{}\t{}".format(country, confirmed))

**#itemmap.py (for example dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 6:

date, time, location, itemtype, amount, cardtype = data

print("{}\t{}".format(itemtype, amount))

**#itemmap.py (for german credit dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 3:

credibility, creditamount, durationofcredit = data

print("{}\t{}".format(credibility, creditamount))

**#itemred.py:**

import fileinput

transaction\_count = 0

sales\_total = 0

for line in fileinput.input():

try:

data = line.strip().split("\t")

if len(data) != 2:

continue

except ValueError:

continue

current\_key, current\_value = data

try:

sales\_total += float(current\_value)

transaction\_count += 1

except ValueError:

continue

print("{}\t{}".format(transaction\_count, sales\_total))

**Output:**

A screenshot of a computer

Description automatically generated with medium confidence

**4) Write a mapper and reducer program for word count by defining a seprator instead of using “\t”**

**#sepmap.py**

import sys

def read\_input(file):

for line in file:

yield line.strip().split(',')

def main(separator="\t"):

data = read\_input(sys.stdin)

for words in data:

for word in words:

print("%s%s%d"%(word, separator, 1))

if \_\_name\_\_ == '\_\_main\_\_':

sep = sys.argv[1]

main(separator=sep)

**#sepred.py:**

import sys

from itertools import groupby

from operator import itemgetter

def read\_mapper\_output(file, separator='\t'):

for line in file:

yield line.rstrip().split(separator, 1)

def main(separator="\t"):

data = read\_mapper\_output(sys.stdin, separator=separator)

for current\_word, group in groupby(data, itemgetter(0)):

try:

total\_count = sum(int(count) for current\_word, count in group)

print("%s%s%d"%(current\_word, separator, total\_count))

except ValueError:

pass

if \_\_name\_\_ == '\_\_main\_\_':

sep = sys.argv[1]

main(separator=sep)

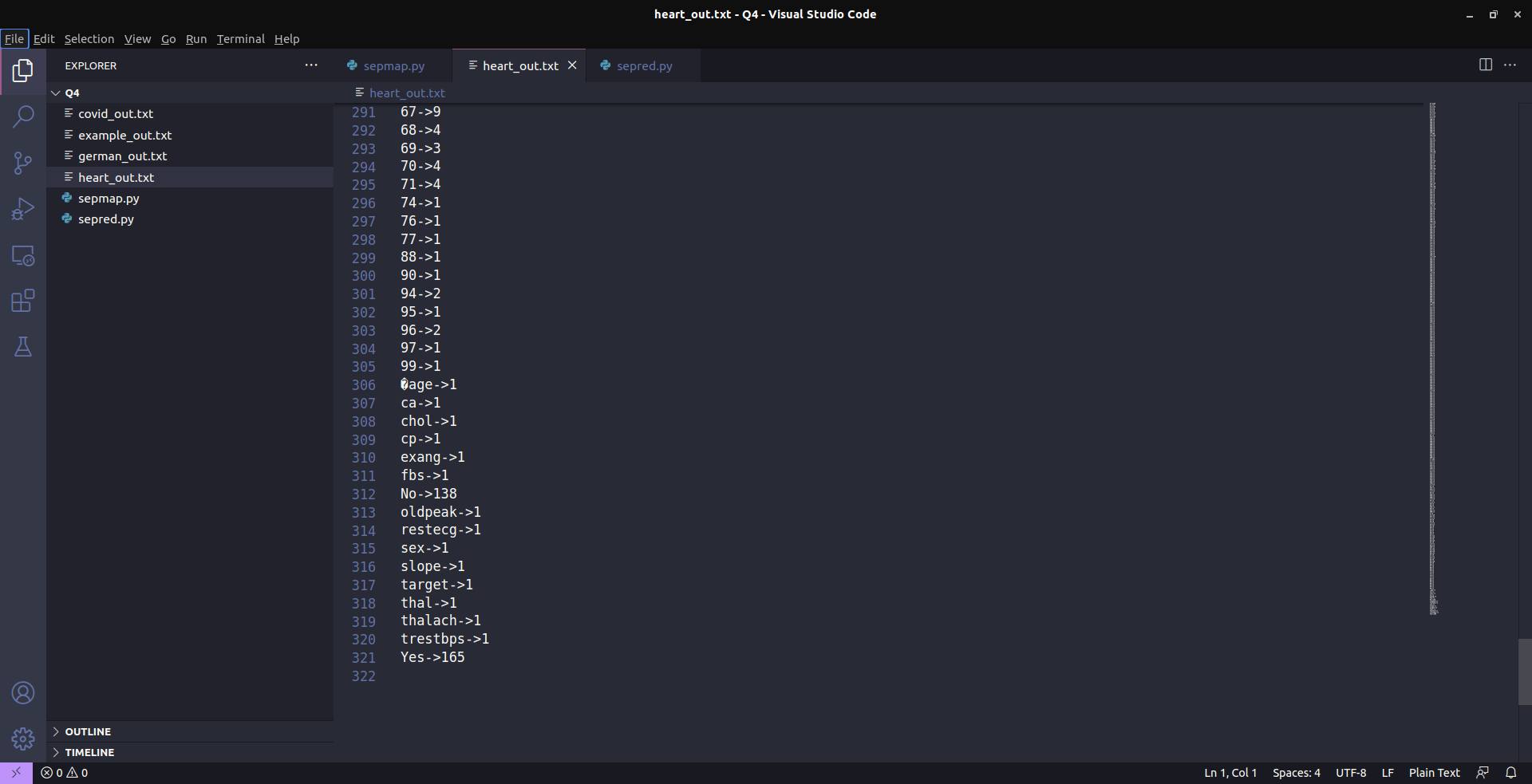
**Output:**

Graphical user interface, text

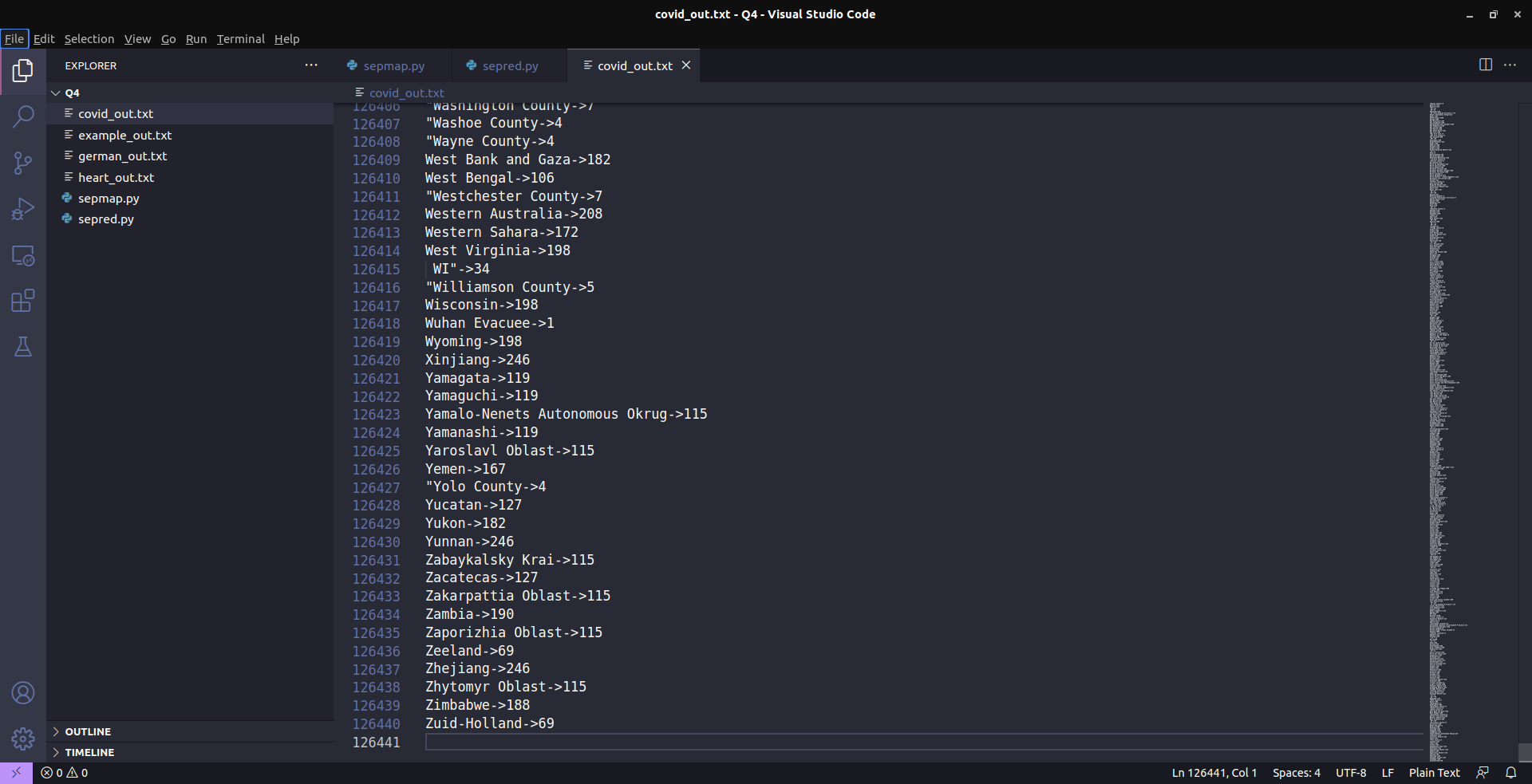
Description automatically generated

**Output files (truncated):**

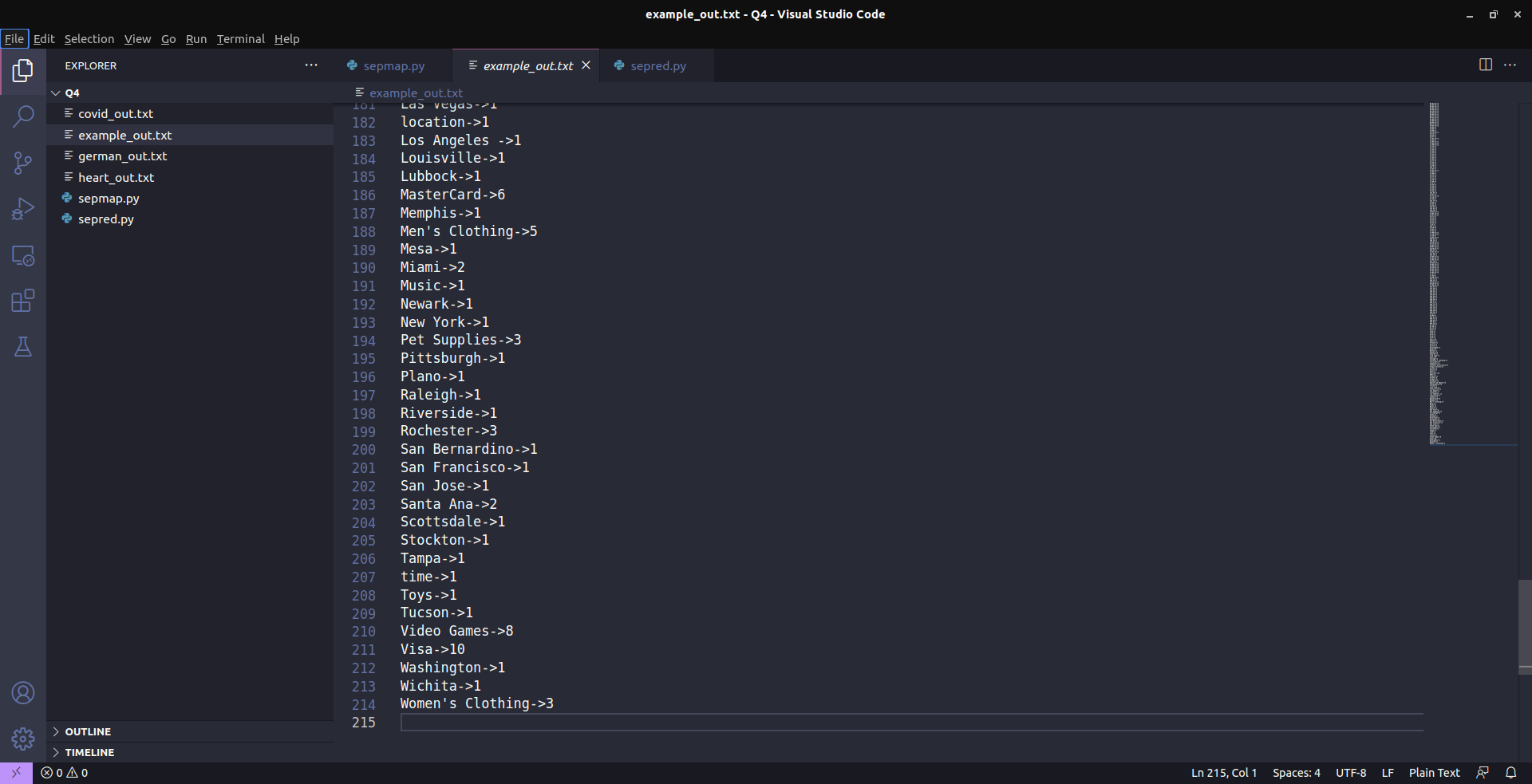
**heart\_out.txt:**

****

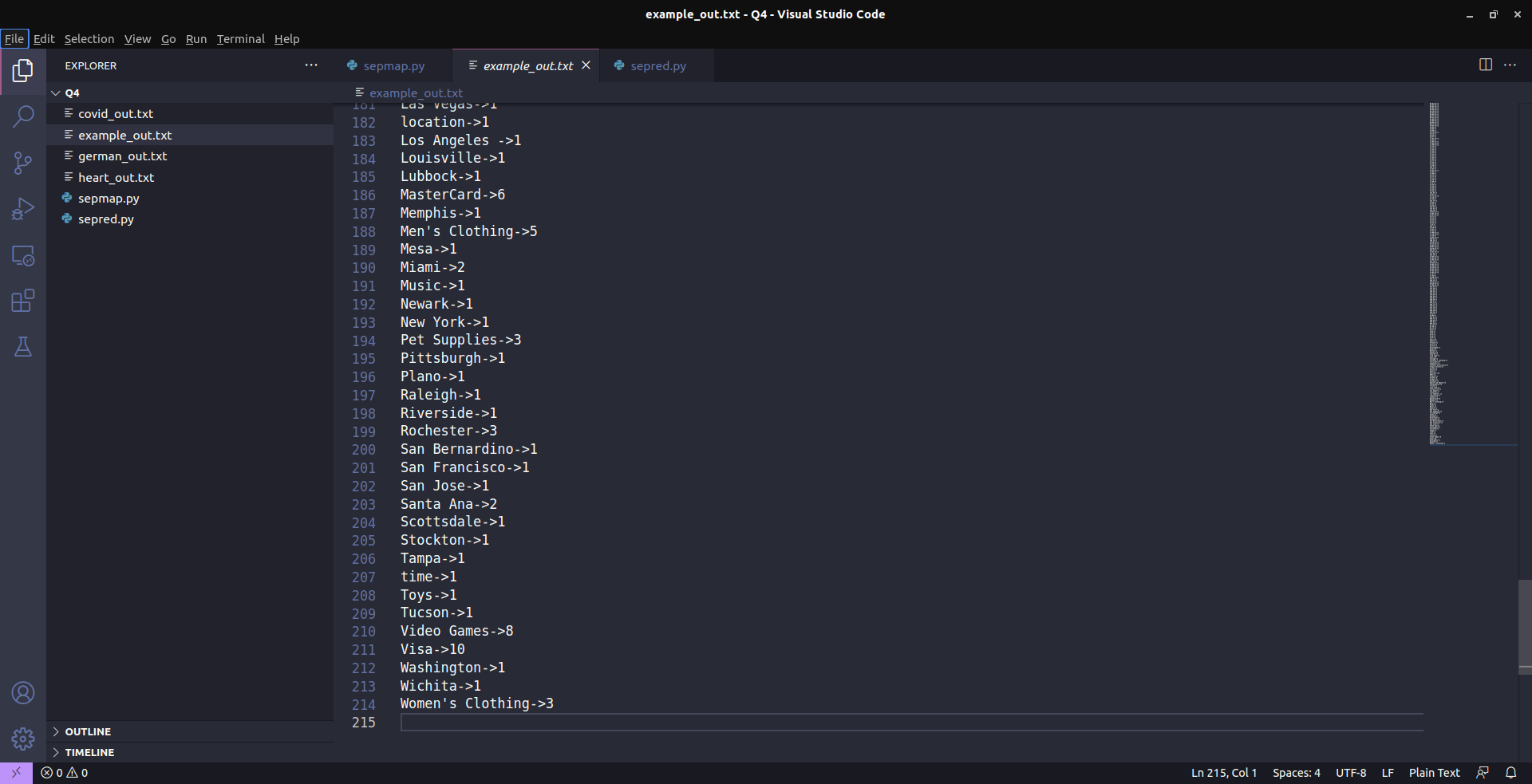
**covid\_out.txt:**

****

**example\_out.txt:**

****

**german\_out.txt:**

****

**5) Try to apply finding max value using map redeuce concept for the output of heart disease dataset, covid 19 dataset, example dataset, german credit dataset**

**#costmap.py (for heart disease dataset):**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 14:

age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, target = data

print("{}\t{}".format(sex, chol))

**#costmap.py (for covid 19 dataset):**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 8:

sno, observationdate, province, country, lastupdate, confirmed, deaths, recovered = data

print("{}\t{}".format(observationdate, confirmed))

**#costmap.py (for example dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 6:

date, time, location, itemtype, amount, cardtype = data

print("{}\t{}".format(itemtype, amount))

**#costmap.py (for german credit dataset)**

import fileinput

for line in fileinput.input():

data = line.strip().split(",")

if len(data) == 3:

credibility, creditamount, durationofcredit = data

print("{}\t{}".format(credibility, creditamount))

**#costred.py:**

import fileinput

max\_val = 0

old\_key = None

for line in fileinput.input():

data = line.strip().split("\t")

if len(data) != 2:

continue

current\_key, current\_value = data

try:

v = float(current\_value)

except ValueError:

continue

if old\_key and (old\_key != current\_key):

print("{}\t{}".format(old\_key, max\_val))

old\_key = current\_key

max\_val = 0

old\_key = current\_key

if float(current\_value) > float(max\_val):

max\_val = float(current\_value)

if old\_key != None:

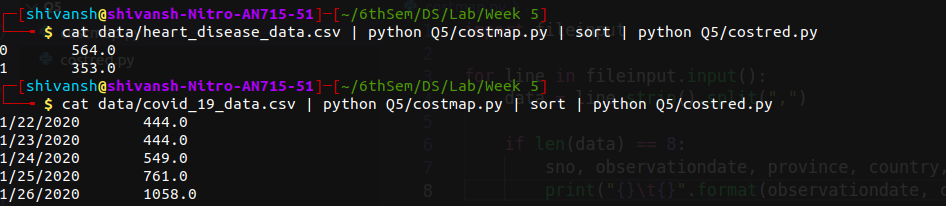
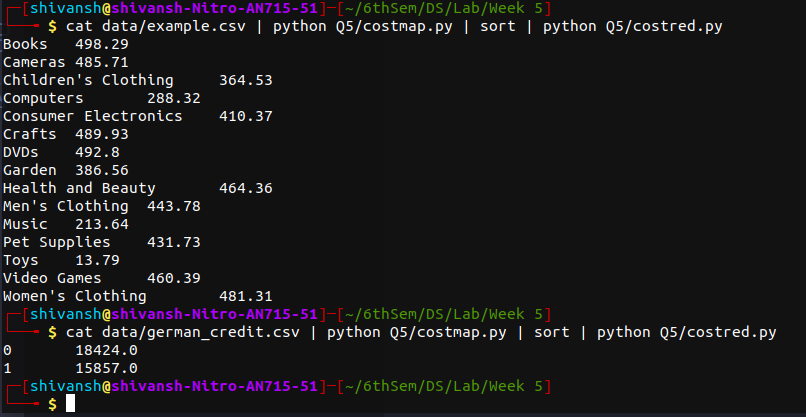
print("{}\t{}".format(old\_key, max\_val))

**Output:**

Graphical user interface, text

Description automatically generated

**Heart\_out.txt example\_out.txt**

****

**covid\_out.txt**

**Text

Description automatically generated**

**German\_out.txt**

**Text

Description automatically generated**

**6) (Instructed to not do)**

**7) Write a map reduce program to count even or odd numbers in randomly generated natural numbers**

**#mapper.py:**

import sys

for line in sys.stdin:

words = line.strip().split()

for word in words:

num = int(word)

if num % 2 == 0:

print("%s\t%d"%("even", 1))

else:

print("%s\t%d"%("odd", 1))

**#reduce.py**

from operator import itemgetter

import sys

current\_word = None

current\_count = 0

word = None

for line in sys.stdin:

try:

word, count = line.strip().split('\t', 1)

count = int(count)

except ValueError:

continue

if current\_word == word:

current\_count += count

else:

if current\_word:

print("%s\t%d"%(current\_word, current\_count))

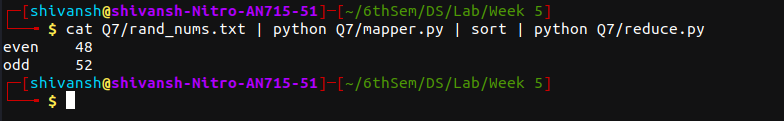
current\_count = count

current\_word = word

if current\_word == word:

print("%s\t%d"%(current\_word, current\_count))

**Output:**

****